

### IN THE CLAIMS:

Please amend claims 1 as follows.

1. (Currently Amended) A method for recovering a database provided with disk back-up, the method comprising the steps of:

- maintaining in a central memory a database comprising a first generation and at least one mature generation, said generations containing memory cells in which data and additionally pointers constituting references between memory cells are stored,

- in the area of mature generations in the central memory, maintaining generation-specific remembered sets, in which ~~the~~ addresses of the pointers pointing to each generation in question are listed,

- allocating memory for the use of an application from the area of the first generation in the central memory,

- in the area of the first generation, periodically collecting live memory cells as a new mature generation into the central memory into the area of mature generations,

- performing garbage collection generationally in the area of mature generations, whereby in connection with the collection a remembered set is examined and live memory cells are copied in the order indicated by the remembered set into a temporally more recent mature generation,

- storing the mature generation after the garbage collection in a separate disk memory,

- as the garbage collection proceeds, making changes, in the area of mature generations, to the references between generations in a generation that has already been stored in disk memory,

wherein

making at least some of said changes in the central memory only,

maintaining in the disk memory, in addition to the most current versions ( $\mathcal{D}'$ ) of the mature generation stored on disk, the previous version ( $\mathcal{D}$ ) stored on disk, and

performing recovery by means of said versions (a) by reconstructing the remembered set of previous version stored on disk, said remembered set indicating the pointers referring to the generation in question, and (b) by changing the pointers indicated by the remembered set to refer to the memory cells of said most current version.

2. (Currently Amended) A method as claimed in claim 1, wherein connection with the recovery

an auxiliary pointer ( $\mathcal{PT}$ ) is set to point to the first memory cell of said most current version ( $\mathcal{D}'$ ),

the remembered set is examined in the same order as in garbage collection, wherein (i) when in connection with examining the remembered set a given memory cell in said previous version ( $\mathcal{D}$ ) is accessed for the first time, a pointer is constructed to point from the memory location corresponding to said memory cell to the memory cell pointed to by the auxiliary pointer, the pointer indicated by the remembered set is changed to

point to the memory cell pointed to by the auxiliary pointer, and the auxiliary pointer is moved one memory cell forward in said most current version, and (ii) when in connection with examining the remembered set a given memory cell is again accessed, the value of the pointer already located at the memory location corresponding to said memory cell is given to the pointer indicated by the remembered set and the moving of the auxiliary pointer is omitted, and

all memory cells in the collected generation ( $\mathcal{D}^2$ ) are examined and remembered sets of older uncollected generations are updated.

3. (Currently Amended) A method as claimed in claim 2, wherein when in connection with examining a remembered set a given memory cell in said previous version ( $\mathcal{D}$ ) is accessed for the first time, said pointer is written directly into said memory cell, as a result of which it points from said memory cell to the memory cell pointed to by the auxiliary pointer, wherein when in connection with examining the remembered set a given memory cell is again accessed, the value of the pointer already located in said memory cell is given to the pointer indicated by the remembered set.

4. (Previously Presented) A method as claimed in claim 1, wherein remembered sets are stored in the central memory in connection with uncollected generations only.

5. (Previously Presented) A method as claimed in claim 4, wherein when it is found in connection with the collection of any generation that a collected memory cell contains a pointer to a memory cell in a generation that has been marked for collection, the address of said pointer is added to the remembered set of said generation.

6. (Previously Presented) A method as claimed in claim 1, wherein garbage collection in the area of mature generations is performed starting from younger generations toward older generations.

7. (Previously Presented) A method as claimed in claim 6, wherein in collecting a mature generation

a new version of the generation is allocated into the area of mature generations, the remembered set of the generation to be collected is examined in such a way that the memory cells pointed to by the pointers indicated by the remembered set are copied into said new version in the order of examining the remembered set, and

the root block of the memory is examined in such a way that the memory cells to which the root block has pointers are copied into said new version.

8. (Previously Presented) A method as claimed in claim 7, wherein when it is found in connection with the collection of a mature generation that a copied memory cell

contains a pointer to a memory cell in a generation marked for collection, the address of said pointer is added to the remembered set of said generation.

9. (Previously Presented) A method as claimed in claim 8, wherein during garbage collection, two generations are combined into one generation if their combined size is smaller than a specific predetermined limit.

Please add new claim 10 as follows:

10. (New) A system for recovering a database provided with disk back-up, the system comprising a central memory and a separate disk memory and being configured to:

maintain in the central memory a database comprising a first generation and at least one mature generation, said generations containing memory cells in which data and additionally pointers constituting references between memory cells are stored,

maintain, in the area of mature generations in the central memory, generation-specific remembered sets, in which addresses of the pointers pointing to each generation in question are listed,

allocate memory for the use of an application from the area of the first generation in the central memory,

periodically collect, in the area of the first generation, live memory cells as a new mature generation into the central memory into the area of mature generations,

perform garbage collection generationally in the area of mature generations, whereby in connection with the collection a remembered set is examined and live memory cells are copied in the order indicated by the remembered set into a temporally more recent mature generation,

store the mature generation after the garbage collection in the separate disk memory, and

as the garbage collection proceeds, make changes, in the area of mature generations, to the references between generations in a generation that has already been stored in disk memory,

wherein the system is configured to

make at least some of said changes in the central memory only,

maintain in the disk memory, in addition to the most current versions of the mature generation stored on disk, the previous version stored on disk, and

perform recovery by means of said versions (a) by reconstructing the remembered set of previous version stored on disk, said remembered set indicating the pointers referring to the generation in question, and (b) by changing the pointers indicated by the remembered set to refer to the memory cells of said most current version.